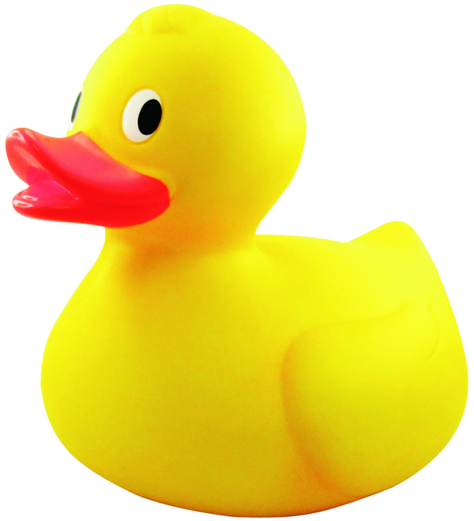
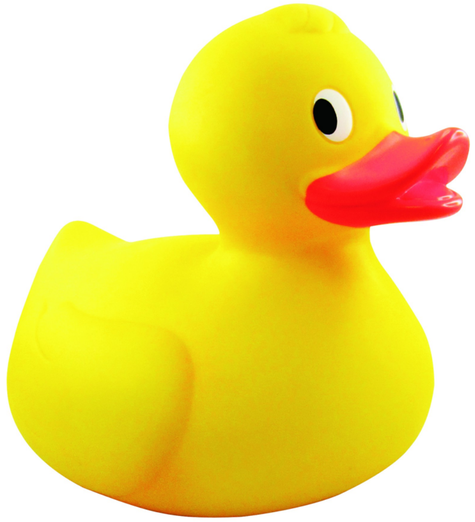
Итераторы



Duck typing



Контейнер поддерживает итерирование, если

```
container.__iter__()
```

Итератор

```
iterator.__iter__()
```

```
iterator.__next__()
```

Итератор

```
iter(iterator)
```

```
next(iterator)
```

Итератор

```
iterator.__iter__()
```

Возвращает сам итератор

```
iterator.__next__()
```

Возвращает очередной элемент,
либо выбрасывает исключение **StopIteration**

Устройство цикла **for**

```
product = 1
for i in [1, 2, 4, 8]:
    product *= i
print(product)
```

```
product = 1
i = iter([1, 2, 4, 8])
while True:
    try:
        product *= next(i)
    except StopIteration:
        break
print(product)
```

Генераторы

Генератор

```
generator.__iter__()
```

```
generator.__next__()
```

```
generator.send(arg)
```

```
generator.throw(typ[,val[,tb]])
```

```
generator.close()
```

Генераторные функции

yield from

Генераторные выражения

Comprehensions

itertools